Probabilistic Inference Models II

ITI0210, lecture 11 (2021)

Review

Inference in general:

Joint Probability Distribution (JPD)

	toothache		¬ toothache	
	catch	¬ catch	catch	¬ catch
cavity	.108	.012	.072	.008
¬ cavity	.016	.064	.144	.576

Pros: can model anything

Cons: high space and time complexity

In practice, independence/known structure allows simplified models

Today

Case study:

Hidden Markov Model (HMM):

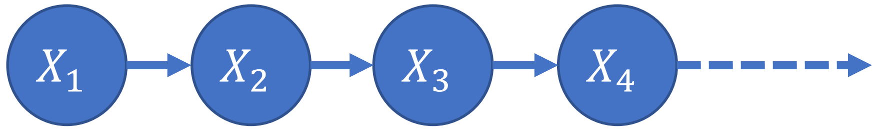
probabilistic model of a sequential process

Markov Chain

Model of a Sequential Process

Markov Chain

Sequence of states in time
$t = 1,2, \ldots$



Markov assumption: each state depends on a short history of earlier states

States and transition probabilities

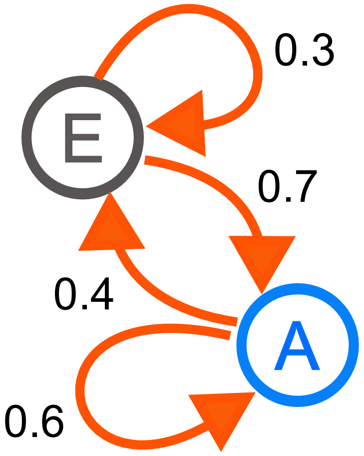
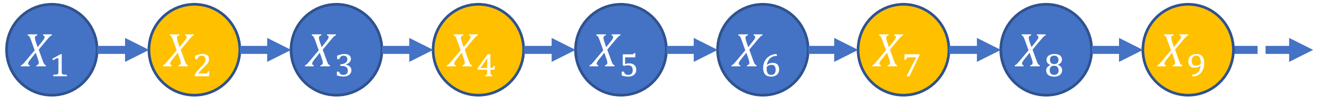


Image: Wikimedia Commons

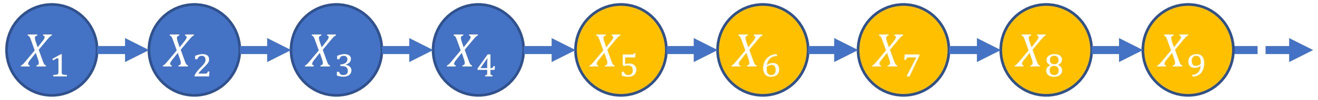
Examples of Markov Chains

High transition probability



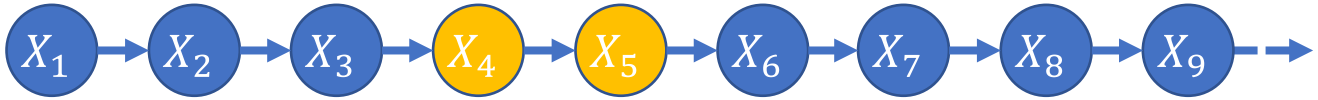
kids taking turns in hide and seek

Low transition probability



Magnetic North Pole

Dominant state



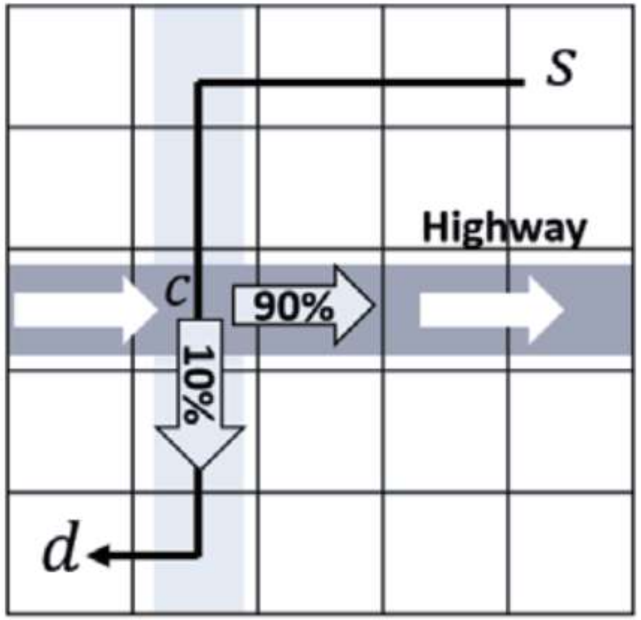
Winner of Bundesliga

in football

Markov Chain

Predicting movement on a city grid

(an example of failure of Markov assumption)



Vahedian, Amin, et al. "Forecasting gathering events through continuous destination prediction on big trajectory data." *SIGSPATIAL* 2017.

Hidden Markov Model

Basic Example

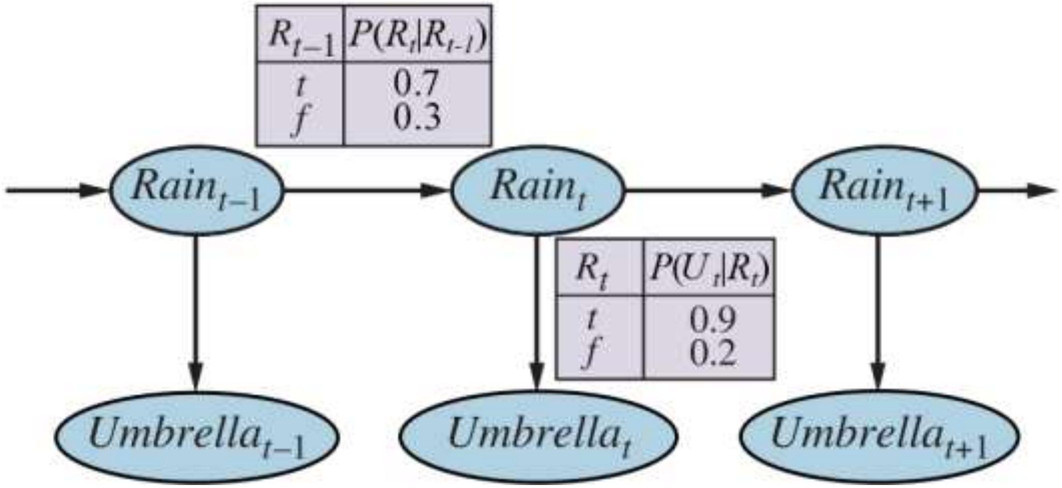
In a HMM, the "state" of Markov Chain is hidden

We see something else **caused** by the state

The "Underground Facility Security Guard" example:

Can't see: outside weather

Can see: boss brought an umbrella



The Math of HMM

State is a Markov Chain (we'll assume 1st order chain for simplicity)

Previous states were X_1 to X_{t-1}, observations E_1 to E_{t-1}

The **current state** $(Rain)$

$$P(X_t|X_{0:t-1}) = P(X_t|X_{t-1})$$

depends **only** on previous state

The observed part $(Umbrella)$, or **emission**:

$$P(E_t|X_{0:t}, E_{1:t-1}) = P(E_t|X_t)$$

depends **only** on current state

The Math of HMM

State is a Markov Chain (we'll assume 1st order chain for simplicity)

Previous states were X_1 to X_{t-1}, observations E_1 to E_{t-1}

The **current state** ($Rain$)

$$P(X_t|X_{0:t-1}) = P(X_t|X_{t-1})$$

X_0 is the assumed state before the chain begins

The observed part ($Umbrella$), or **emission**:

$$P(E_t|X_{0:t}, E_{1:t-1}) = P(E_t|X_t)$$

Demo

Let’s play a board game…

… and cheat! We have two dice, “fair” and a “cheat” die

Can sometimes swap a die (low transition probability)

http://lambda.ee/wiki/Iti0210w101demo

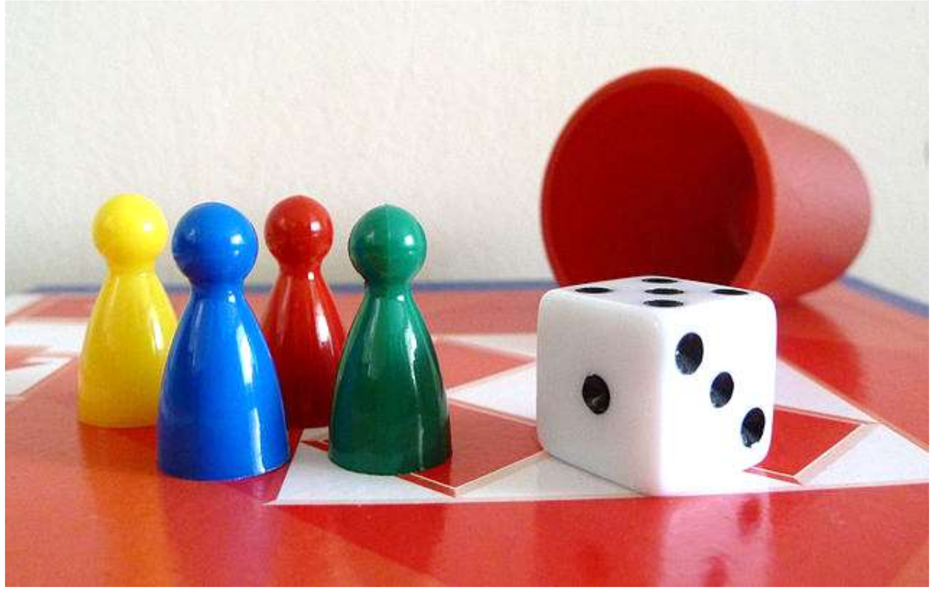


Image: Wikimedia Commons

Inferring Hidden States

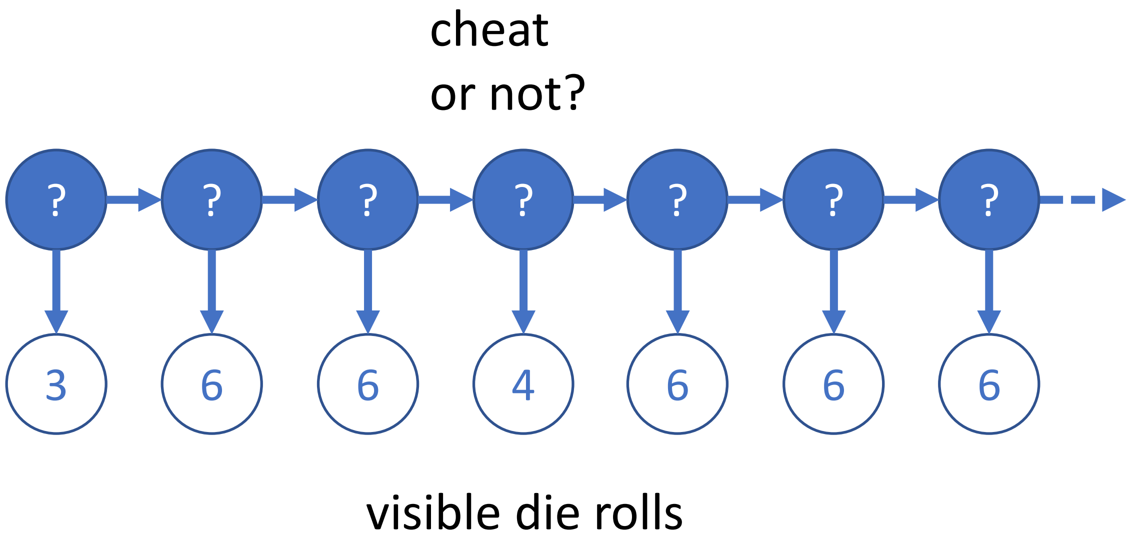
The Message Passing algorithm

Inference Problem

Inference task:

- emissions known
- probabilities known

Find the hidden states



Inferring the States

Straight calculation over JPD:

for 50 die rolls, 2^{50} or 1126 trillion calculations

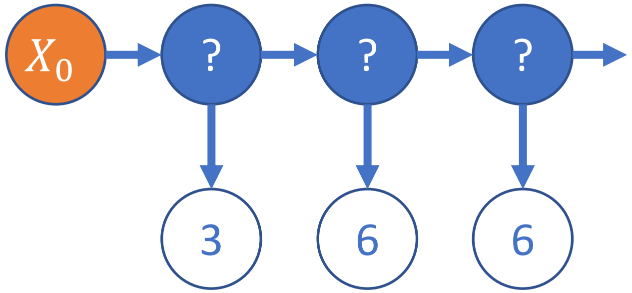
(50 hidden variables with 2 values)

Exploit the conditional independence structure

(e.g. emission **independent** of everything else, **given** current state)

Forward Inference

Start simple:



what is the internal state before the chain begins?

Answer:

$$P(X_0)$$

Since we don't actually know, let's use a distribution:

"fair" : 0.5, "cheat" : 0.5

Forward Inference

Previous state:

“fair” : 0.5, “cheat” : 0.5

Let’s call this a **forward message**

$$f_0 = (0.5, 0.5)$$

and:

$$f_0(\text{"fair"}) = 0.5$$
$$f_0(\text{"cheat"}) = 0.5$$

Forward Inference

Evidence/
emission,
always fixed

Previous state:

"fair" : 0.5, "cheat" : 0.5

Value we are
computing for;
plug in each of
"fair", "cheat"

Current emission (die roll): 3

Using **both** pieces of info

$$f_k(x_k) = P(e_k|x_k) \sum_{x_{k-1} \in \{fair, cheat\}} P(x_k|x_{k-1}) f_{k-1}(x_{k-1})$$

Derivation in AIMA book, 15.2.1

Forward Inference

Previous state:

"fair" : 0.5, "cheat" : 0.5

Loop over the distribution in previous state (given by the message)

Current emission (die roll): 3

Using both pieces of info

$$f_k(x_k) = P(e_k|x_k) \sum_{x_{k-1} \in \{fair, cheat\}} P(x_k|x_{k-1}) f_{k-1}(x_{k-1})$$

Backward Inference

For any step k, there is also future evidence $E_{k+1:t}$

Backwards message:

$$b_{k+1}(x_k) = \sum_{x_{k+1} \in \{fair, cheat\}} P(e_{k+1}|x_{k+1}) P(x_{k+1}|x_k) b_{k+2}(x_{k+1})$$

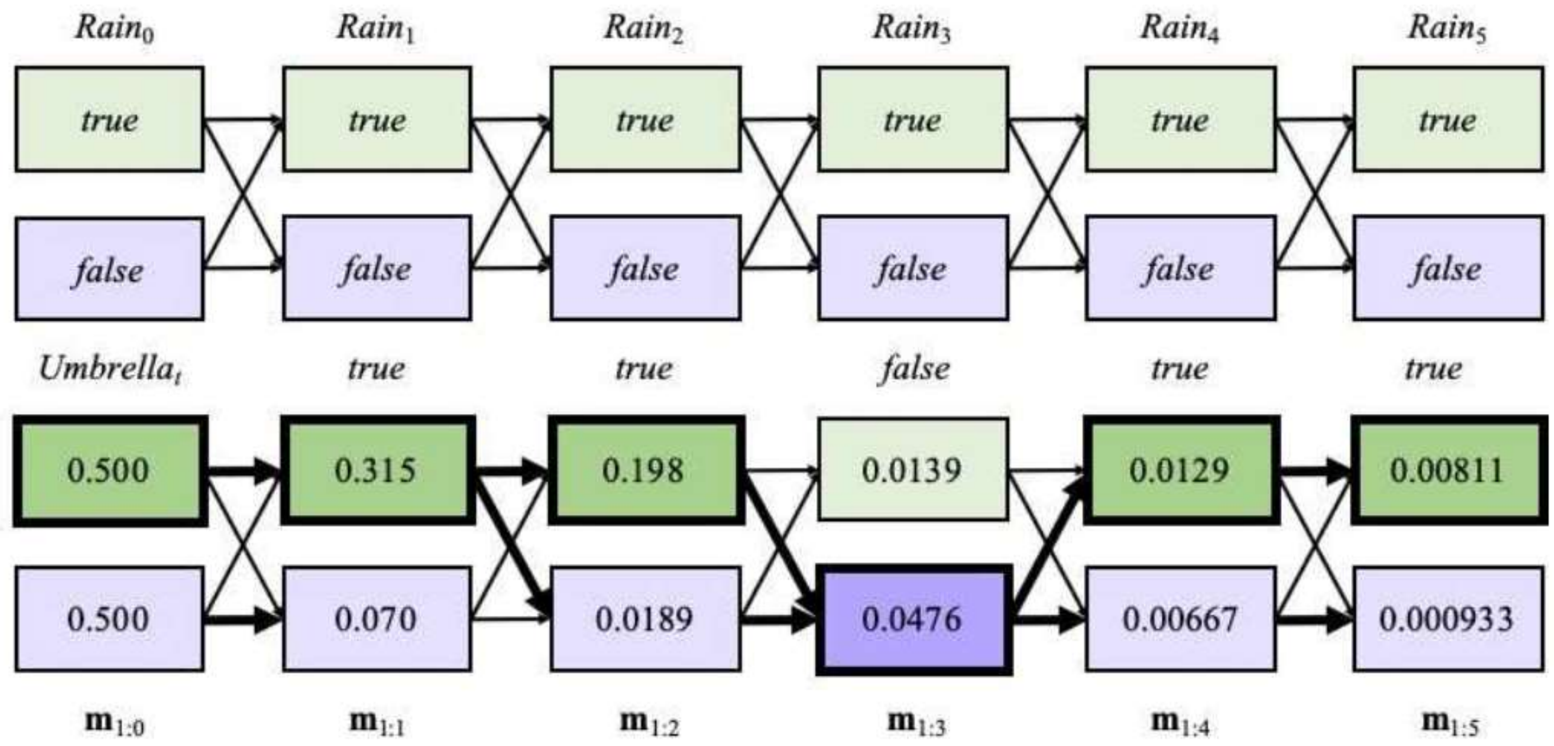
Compute for "fair" and "cheat" in turn

Loop over each value in the previous message

Evidence/ emission, always fixed

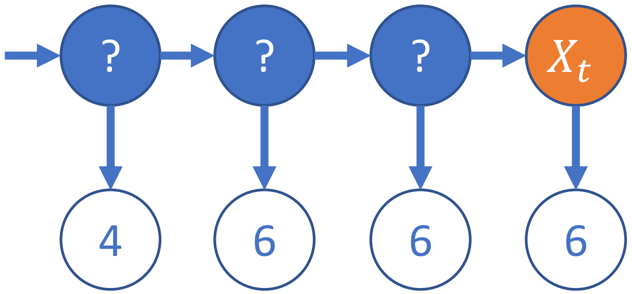
Why Always Sum over States?

Multiple paths enter and leave one state, e.g. $Rain_2$:



The Other End Point

If there are no states past t:



Use **uniform** distribution for backward message

(Book uses all 1-s but this will be normalized anyway)

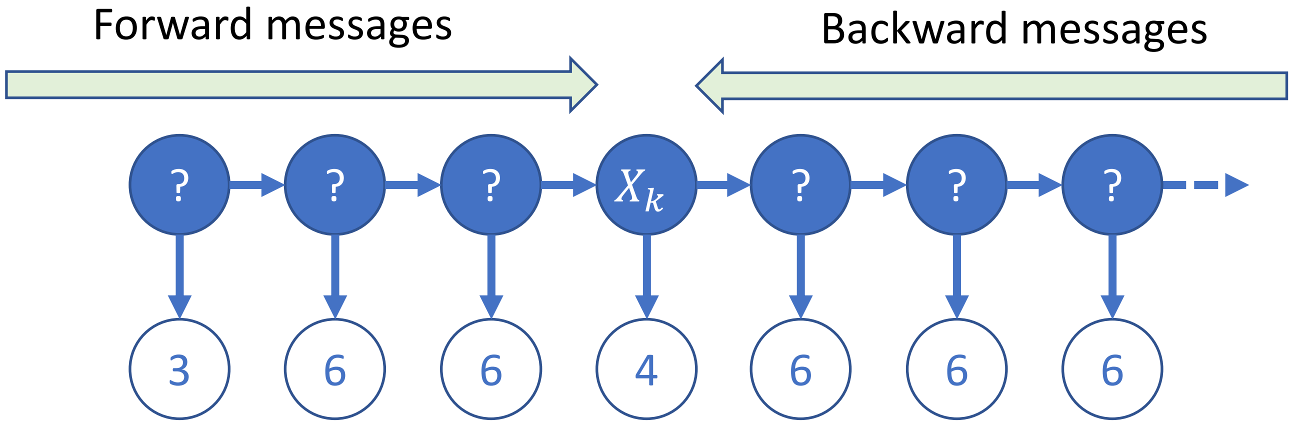
$$b_{t+1}(\text{fair}) = 0.5$$
$$b_{t+1}(\text{cheat}) = 0.5$$

Smoothed Estimate

For time step k

$$P(X_k = \text{cheat}) = \alpha f_k(\text{cheat}) b_{k+1}(cheat)$$
$$P(X_k = \text{fair}) = \alpha f_k(\text{fair}) b_{k+1}(fair)$$

α – choose a number such that the sum of probabilities over all states is 1



Smoothed Estimate

```
def smoothed_estimate(E, t=50):
    f_prev = (prior["fair"], prior["cheat"])
    f_X = []
    for k in range(t):
        f_k = forward(f_prev, E[k])
        f_X.append(f_k)
        f_prev = f_k

    b = (0.5, 0.5)
    smooth_X = []
    for k in range(t-1, -1, -1):
        smooth_X.append(normalize([f_X[k][0] * b[0], f_X[k][1] * b[1]]))
        b = backward(b, E[k])
    smooth_X.reverse()
    return smooth_X
```

Forward messages for each time step

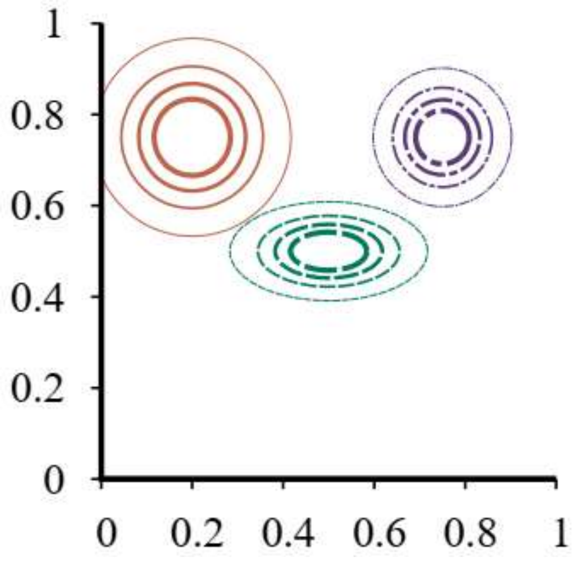
Forward messages meet backward messages

Learning Distributions

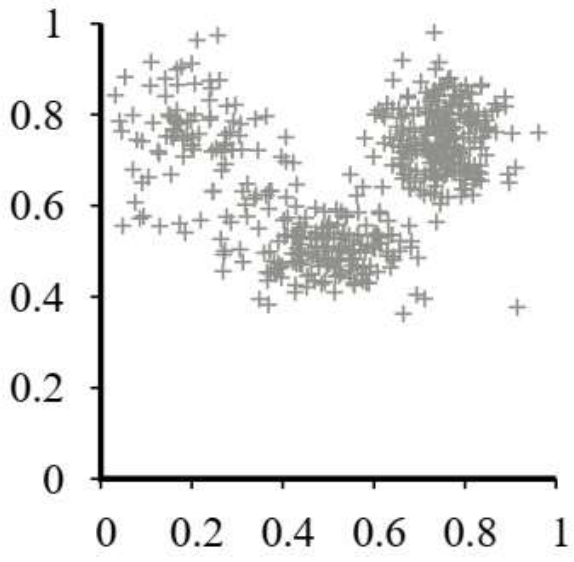
What if the distribution is not given?

EM algorithm

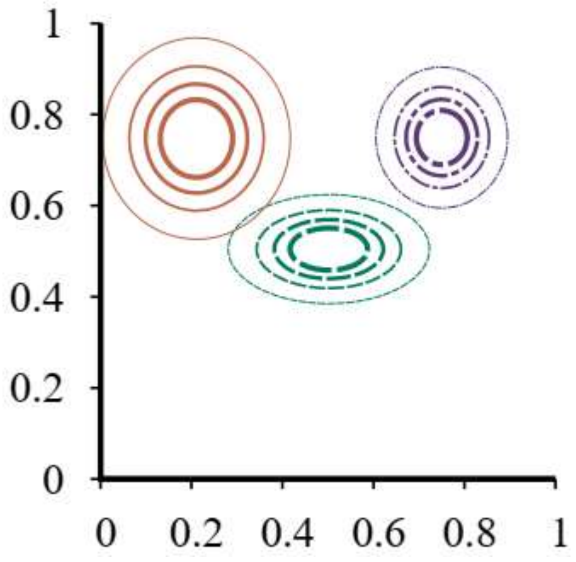
Data itself can reveal the distribution:



original Gaussians (models e.g. the probability of a dart throw)



sampled data



Gaussians learned with the EM algorithm

EM algorithm idea

Don’t know a real distribution $P(X)$

Let’s guess the distribution is some $\pi(X)$ (may initialize randomly)

E-step:

Calculate the expected values of hidden variables, given $\pi(X)$

M-step:

Update $\pi(X)$, using the expected values as if they were real data

Repeat E-step and M-step until convergence

EM for our Hidden Markov Model

Assume we don't know the transition probabilities $P(X_{k+1}|X_k)$

E-step:

Use our `smoothed_estimate()` with some guess $\pi(X_{k+1}|X_k)$

M-step

Update $\pi(X_{k+1}|X_k)$ using the Markov chain of inferred hidden states

M-step for HMM

Inferred hidden states:

	X_1	X_2	X_3	X_4
cheat	0.32	0.75	0.8	0.99
fair	0.68	0.25	0.2	0.01

Count step $k = 1$ as: "cheat" occurred 0.32 times, "fair" 0.68 times

$$\pi(X_{k+1} = fair | X_k = fair) = \frac{(0.68 \times 0.25) + (0.25 \times 0.2) + (0.2 \times 0.01)}{0.68 + 0.25 + 0.2} = 0.196$$